

A

Docket No. 004066/CONS/MBE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application
Patent Commissioner for Patents
Washington, D.C. 20231

Re: Inventor(s): John F. Arackaparambil, Tom Chi, Billy Chow, Patrick M. D'Souza, Parris Hawkins, Charles Huang,
Jett Jensen, Badri N. Krishnamurthy, Pradeep M. Kulkarni, Prakash M. Kulkarni, Wen Fong Lin,
Shantha Mohan, Bishnu Nandy, Huey-Shin Yuan

Jc530 U.S. PTO
09/363966
07/29/99

Title: COMPUTER INTEGRATED MANUFACTURING TECHNIQUES

Transmitted herewith is the patent application identified above, including:

- ☒ Specification, claims and abstract, totaling 41 pages.
- ☒ Drawings totaling 10 pages, Formal ☒ Informal.
- ☐ Executed Declaration and Power of Attorney.
- ☐ Assignment of the invention to Applied Materials, Inc.
- ☐ Assignment Recordation Cover Sheet

FEE CALCULATION					
Fee Items	Claims Filed	Included With Basic Fee	Extra Claims	Fee Rate	Total
Total Claims	69	- 20 =	49	X \$18.00	882.00
Independent Claims	10	- 3 =	7	X \$78.00	546.00
Basic Filing Fee				\$760.00	\$760.00
TOTAL FEES					\$2,188.00

- ☒ The Commissioner is hereby authorized to charge \$2,188.00 to Deposit Account No. 01-1651.
- ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 01-1651. A duplicate copy of this transmittal is enclosed.
- ☒ Please address all future correspondence to:

PATENT COUNSEL
APPLIED MATERIALS, INC.
Legal Affairs Department
P.O.BOX 450A
Santa Clara, CA. 95052

I hereby certify that this correspondence is being deposited with the United States Postal Service as express mail in an envelope addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Express Mail Receipt No

EL26633798145

Date of Deposit

7/29/99

Signature

Beel Mulcahy

Respectfully submitted,

Robert W. Mulcahy
Registration No. 25,436

COMPUTER INTEGRATED MANUFACTURING TECHNIQUES

GOVERNMENT RIGHTS IN THE INVENTION

5 This invention was made with United States Government support under Cooperative Agreement No. 70NANB7H3043 awarded by NIST. The United States Government has certain rights in the invention.

FIELD OF THE INVENTION

10 The present invention relates to techniques for computer integrated manufacturing.

BACKGROUND OF THE INVENTION

15 A semiconductor device such as an IC (integrated circuit) generally has electronic circuit elements such as transistors, diodes and resistors fabricated integrally on a single body of semiconductor material. The various circuit elements are connected through conductive connectors to form a complete circuit which can contain millions of individual circuit elements. Integrated circuits are typically fabricated from semiconductor wafers in
20 a process consisting of a sequence of processing steps. This process, usually referred to as wafer fabrication or wafer fab, includes such operations as oxidation, etch mask preparation, etching, material deposition, planarization and cleaning.

25 A summary of an aluminum gate PMOS (p-channel metal oxide semiconductor transistor) wafer fab process 40 is schematically shown in FIG. 1, illustrating major processing steps 41 through 73, as described in W.R. Runyan et al., Semiconductor Integrated Circuit Processing Technology, Addison-Wesley Publ. Comp. Inc., p.48, 1994. Each of these major processing steps typically include several sub steps. For example, a major processing step such as metallization to provide an aluminum layer by means of
30 sputter deposition in a wafer fab chamber is disclosed in U.S. Pat. No. 5,108,570 (R.C. Wang, 1992). This sputter deposition process is schematically shown in sub steps 81 through 97 of process 80, see FIG. 2.

FIGS. 1 and 2 show sequential wafer fab processes. It is also known to utilize wafer fab sub systems which provide parallel processing steps. Such sub systems typically include one or more cluster tools. A cluster tool as defined herein includes a system of chambers and wafer handling equipment wherein wafers are processed in the cluster tool chambers without leaving a controlled cluster tool environment such as vacuum. An example of a cluster tool is disclosed in U.S. Pat. No. 5,236,868 (J. Nulman, 1993) which employs a vacuum apparatus having a central chamber and four processing chambers. A wafer handling robot in the central chamber has access to the interior of each the processing chambers in order to transfer wafers from the central chamber into each of the chambers while keeping the wafers in a vacuum environment. In one example, wafers in the '868 cluster are first transferred for processing to a cleaning chamber, then to a PVD (physical vapor deposition) chamber, followed by transfer to an annealing chamber and subsequently to a degassing chamber, thus utilizing a sequential process. It is also known to use cluster tools such as those disclosed in the '868 patent to process wafers in chambers which are used in parallel. For example, if a slow processing step is followed by a fast processing step, three chambers can be used in parallel for the slow process while the fourth chamber is used for the fast process.

Effective wafer inventory management is necessary for maintaining inventories of unprocessed or partly processed wafers at a minimum and thereby minimizing the unit cost of the semiconductor devices which are produced in the wafer fab. Minimizing inventories of wafers in process also has a wafer yield benefit because it is well known that the longer wafers are in the process, the lower their yield. Wafer inventory management typically uses scheduling techniques to maximize equipment capabilities in view of the demand for processed wafers, for example by scheduling parallel and series processing steps to avoid processing bottlenecks. It is well known to those of ordinary skill in the art that in-process wafer inventory management is facilitated by in-process wafer tracking, such as tracking wafer lots and wafer cassettes throughout a wafer fab. Effective inventory management of a wafer fab also requires a low incidence of bottlenecks or interruptions due to unscheduled down times which can for example be caused by unscheduled maintenance, interruptions resulting from processing parameters which are outside their specified limits, unavailability of required materials such as a process gas, unavailability of necessary maintenance replacement parts or unavailability

of a processing tool such as a chamber.

Many components or sub-systems of a wafer fab are automated in order to achieve a high degree of processing reliability and reproducibility and to maximize yields. Wafer fab tools such as chambers are typically controlled by a computer using a set of instructions which are generally known as a recipe for operating the process which is executed by the tool. However, it is recognized that a high degree of automation wherein various processes and metrologies are integrated, is difficult to achieve due to the complexity and inter dependency of many of the wafer fab processes, see for example Peter van Zandt, Microchip Fabrication, 3rd ed., McGraw-Hill, pp. 472 – 478, 1997. Manufacturing systems such as a wafer fab are known to utilize software which provides an MES (manufacturing executions systems) function. Desirably, a wafer fab MES should be integrated for an entire wafer fab in order to achieve centralized wafer fab management and control. However, it is well known to those of ordinary skill in the art that a commercial wafer fab typically includes semiconductor processing tools from different equipment manufacturers, resulting in tool compatibility difficulties when attempts are made to develop an integrated MES. Another shortcoming of currently available wafer fab MES is the need for extensive software programming for each process change in the wafer fab, such as is necessary for changing a recipe, adding or replacing a tool, or changing the wafer fab to make a different wafer product.

It is well known to those of ordinary skill in the art that the functions of semiconductor manufacturing equipment, including for example a wafer fab, can be defined in basic equipment states such as the six states schematically illustrated in FIG. 3, see SEMI E10-96, Standard For Definition And Measurement Of Equipment Reliability, Availability, And Maintainability (RAM), published by Semiconductor Equipment and Materials International (SEMI), pp. 1-23, 1996. The semiconductor industry typically uses these six equipment states to measure and express equipment RAM (reliability availability and maintainability), based on functional equipment issues which are independent of who performs the function. These six basic equipment states include non-scheduled time (FIG. 3), unscheduled downtime 104, scheduled downtime 106, engineering time 108, standby time 110 and productive time 112. Non-scheduled time 102 represents the time period wherein the equipment is not scheduled to be used, for example unworked shift.

Unscheduled downtime 104 concerns time periods wherein the equipment is not in a condition to perform its intended function, e.g. during equipment repair. Scheduled downtime 106 occurs when the equipment is capable of performing its function but is not available to do this, such as process setup or preventive maintenance. Engineering time 108 concerns the time period wherein the equipment is operated to conduct engineering tests, for example equipment evaluation. Standby time 110 is a time period wherein the equipment is not operated even though it is in a condition to perform its intended function and is capable of performing its function, for example no operator is available or there is no input from the relevant information systems. Productive state 112 represents the time period wherein the equipment is performing its intended function, such as regular production and rework.

Total time period 114, see FIG. 3, is the total time during the period being measured; this includes the six equipment states 102, 104, 106, 108, 110 and 112. Operations time 116 concerns the total time period of states 104, 106, 108, 110 and 112. Operations time 116 includes equipment downtime 118 consisting of states 104 and 106, and equipment uptime 120. Equipment uptime 120 includes engineering time 108 and manufacturing time 122 which consists of standby time 110 and productive time 112.

FIGS. 4 and 5 provide more detailed schematic illustrations of the six equipment states shown in FIG. 3, see SEMI E10-96, at pp. 1-6. As depicted in FIG. 4, total time 114 consists of non-scheduled time 102 and operations time 116. Non-scheduled time 102 includes unworked shifts 130, equipment installation, modification, rebuilding or upgrading 132, off-line training 134 and shutdown or start-up time period 136. Operations time 116, as schematically illustrated in FIG. 5, consists of equipment downtime 118 and equipment uptime 120. Equipment downtime 118 consists of unscheduled downtime 104 and scheduled downtime 106. Unscheduled downtime 104 includes downtime for maintenance delay 140, repair time 142, changing consumables/chemicals 144, out of specification input 146 or facilities related downtime 148. Scheduled downtime 106 concerns downtime for maintenance delay 150, production test 152, preventive maintenance 154, changing consumables/chemicals 156, setup 158 or facilities related 159.

Equipment uptime 120, depicted in FIG. 5, consists of engineering time 108 and manufacturing time 122. Engineering time 108 includes process experiments 160 and equipment experiments 162. Manufacturing time 110 consists of standby time 110 and productive time 112. Standby time 110 includes time during which there is no operator 180, no product 182, no support tool 184 or when an associated cluster module is down 186. Productive time 112 concerns a time period during which there is regular production 190, work for a third party 192, rework 194 or an engineering run 196. The various equipment states as described in connection with FIGS. 3 – 5 provide a basis for communicating and evaluating RAM related equipment information in the semiconductor industry. RAM related equipment information includes topics which are well known to those of ordinary skill in the art such as: equipment reliability, equipment availability, equipment maintainability and equipment utilization, see for example SEMI E10-96 at pp. 6-11. Generally, MES functions can be employed to keep track of information regarding equipment states in manufacturing systems such as a wafer fab.

Advances in semiconductor materials, processing and test techniques have resulted in reducing the overall size of the IC circuit elements, while increasing their number on a single body. This requires a high degree of product and process control for each processing step and for combinations or sequences of processing steps. It is thus necessary to control impurities and particulate contamination in the processing materials such as process gases. Also, it is necessary to control processing parameters such as temperature, pressure, gas flow rates, processing time intervals and input sputter power. As illustrated in FIGS. 1 and 2, a wafer fab includes a complex sequence of processing steps wherein the result of any particular processing step typically is highly dependent on one or more preceding processing steps. For example, if there is an error in the overlay or alignment of etch masks for interconnects in adjacent IC layers, the resulting interconnects are not in their proper design location. This can result in interconnects which are packed too closely, forming electrical short defects between these interconnects. It is also well known that two different processing problems can have a cumulative effect. For example, a misalignment of interconnect etch masks which is not extensive enough to result in an electrical short, can still contribute to causing an electrical short if the process is slightly out of specification for allowing (or not detecting) particulate contamination having a particle size which would not have caused an electrical short if the interconnect masks had

been in good alignment.

Processing and/or materials defects such as described above generally cause a reduced wafer fab yield, wherein the yield is defined as the percentage of acceptable wafers that are produced in a particular fab. In-process tests and monitoring of processing parameters are utilized to determine whether a given in-process product or process problem or defect indicates that intervention in the process run is necessary, such as making a processing adjustment or aborting the run. Consequently, product and process control techniques are used extensively throughout a wafer fab. When possible, yield problems are traced back to specific product or processing problems or defects to ultimately improve the yield of the wafer fab. High yields are desirable for minimizing manufacturing costs for each processed wafer and to maximize the utilization of resources such as electrical power, chemicals and water, while minimizing scrap re-work or disposal.

It is known to use SPC (statistical process control) and SQC (statistical quality control) methods to determine suitable wafer fab control limits and to maintain the process within these limits, see for example R. Zorich, Handbook Of Quality Integrated Circuit Manufacturing, Academic Press Inc., pp. 464-498, 1991. SPC and SQC methodologies suitable for a wafer fab include the use of control charts, see for example R. Zorich at pp. 475-498. As is well known to those of ordinary skill in the art, a control chart is a graphical display of one or more selected process or product variables, such as chamber pressure, which are sampled over time. The target value of a particular variable and its upper and lower control limits are designated on the chart, using well known statistical sampling and computation methods. The process is deemed out of control when the observed value of the variable, or a statistically derived value such as the average of several observed values, is outside the previously determined control limits. Control limits are typically set at a multiple of the standard deviation of the mean of the target value, such as for example 2σ or 3σ . The target value is derived from a test run or a production run which meets such wafer fab design criteria as yield, process control and product quality. SPC and SQC are considered synonymous when used in the above context, see R. Zorich at p. 464.

Accordingly, a need exists for methods and techniques which provide improved computer implemented integration of semiconductor manufacturing techniques in order to optimize process control, quality, yield and cost reduction. Also, there is a need for centralized wafer fab management and control through a computer integrated manufacturing system which facilitates processing or equipment changes without extensive software programming.

SUMMARY OF THE INVENTION

The present invention provides novel techniques for computer integrated manufacturing, particularly for manufacturing integrated circuit structures such as semiconductor wafers. These novel techniques provide the needed improvements in computer integration.

In one embodiment of the present invention a novel factory automation lifecycle is provided which includes SW for lifecycle activities for developing and integrating, installing and administrating, factory modeling, manufacturing planning, manufacturing controlling, monitoring and tracking, and lifecycle activities for analyzing manufacturing results. Output from an analyzing manufacturing results lifecycle activity can provide an input to other lifecycle activities, such as the factory modeling lifecycle activity. Framework components are associated with various lifecycle activities.

In another embodiment of the present invention a novel method for managing a processing system is provided which includes utilizing framework software components, application software components and software building blocks. The application components provide instructions for managing the system while the framework components are employed to manage the application components. The building blocks are adapted for forming or modifying framework and application components. A factory automation lifecycle includes the framework components. A novel tool integration component is employed by the novel method to communicate instructions to processing tools of the system. The tool integration component comprises a tool interface program and a tool integration component adapter. Instructions for managing the system can be modified by inputting data.

In still another embodiment of the present invention a novel method for processing a product includes determining the specifications for processing the product and then managing the process by means of a novel distributed factory system framework which includes framework components, application components and SW building blocks. The novel distributed factory system can be modified, if necessary, by inputting data. Computer implemented instructions for managing are formed by application components. These instructions are communicated to the process for manufacturing a product, for example by utilizing a tool integration component. The instructions are then implemented in the process, for example for fabricating integrated circuit structures.

In yet another embodiment of the present invention a novel method for starting a wafer fab run includes determining the sequence of processing steps and subsequently forming a workflow defining this sequence in a visual workflow component. The visual workflow component is included in a novel distributed factory system framework comprising framework components and application components. A request is then made to the visual workflow component to start the run by means of a work in progress management component or a GUI.

In another embodiment of the present invention an apparatus is provided including product processing equipment, a central processing unit, a link for operably linking the processing equipment to the central processing unit, a memory for storing digitally coded data structures, and data structures comprising a novel factory automation lifecycle. The present embodiment also provides for data structures including application components and building block components.

In still another embodiment of the present invention a distributed factory system framework is provided for managing a processing system, including a digitally coded first data structure comprising framework components, a second data structure including application components and a link for communicating digitally coded instructions to the processing system.

In yet another embodiment of the present invention a novel apparatus is provided comprising processing equipment and a novel distributed factory system framework for managing a processing system.

5 In additional embodiments of the present invention, novel data storage devices are provided comprising data structures such as novel factory automation lifecycle activity data structures, framework component data structures, application component data structures and building block data structures.

10 BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart schematically illustrating a prior art wafer fab process.

FIG. 2 is a flowchart schematically illustrating a prior art wafer fab sputter metallization process.

15 FIG. 3 is a stack chart schematically illustrating prior art equipment time states.

FIG. 4 is a block diagram schematically showing prior art equipment time states of the stack chart illustrated in FIG. 3.

FIG. 5 is a block diagram schematically showing prior art equipment time states of the stack chart illustrated in FIG. 3.

20 FIG. 6 is a block diagram schematically illustrating a factory automation lifecycle of the present invention.

FIG. 7 is a schematic diagram illustrating interactions between framework components of the present invention.

25 FIG. 8 is a block diagram schematically illustrating component servers of the present invention.

FIG. 9 is a block diagram schematically illustrating a tool integration component of the present invention.

FIG. 10 is a schematic diagram illustrating a sequence of messages from a visual workflow component to a tool interface program of the present invention.

30

DETAILED DESCRIPTION OF THE INVENTION

While describing the invention and its embodiments, certain terminology will be utilized for the sake of clarity. It is intended that such terminology includes the recited embodiments as well as all equivalents.

5 In one embodiment of the invention a DFS/F (distributed factory system framework) SW (software) environment is provided to automate, integrate and coordinate factory MES (manufacturing execution system(s)) comprising equipment steps, decision steps and data steps which can be present in a processing, manufacturing or fabricating system or facility such as a wafer fab for processing or fabricating semiconductor structures, such as IC (integrated circuit) structures. The expression "FW" (framework) as defined herein, includes a collection of linked SW structures, components or classes that provide a functionality or a set of services. The expression "MES" as defined herein, includes a collection of SW data structures for starting processing related tasks, managing and/or controlling work in progress and facilitating the use of resources, such as materials, equipment, information and historical data for the execution of processing/manufacturing/fabrication tasks, optionally including testing and data gathering tasks. The expression "IC structures" as defined herein, includes completely formed ICs and partially formed ICs.

20 DFS/F of the present invention includes a novel FALC (factory automation lifecycle) 200, shown in FIG. 6, to form an overall structure for integrated factory automation MES. FALC 200 is adapted for integrating, automating, managing or controlling various manufacturing related aspects of a processing, manufacturing or fabricating system or facility, thereby forming a factory model. These aspects can include wafer fab related equipment, such as wafer fab tools, methods such as wafer fab processing procedures, materials such as wafer fab process gases, inventory control, such as in-process wafer inventory control, work in process status determinations, evaluating in-process test data, monitoring equipment functions and quality management features, such as SPC (statistical process control). The novel lifecycle additionally facilitates the integration of tools, equipment or software from different tool, equipment or software suppliers to provide a coordinated manufacturing or fabricating facility wherein several tools are integrated.

Novel FALC 200, depicted in FIG. 6, includes an SW developing and integrating lifecycle activity 210, an installing and administrating lifecycle activity 220, a factory modeling lifecycle activity 230, a manufacturing planning lifecycle activity 240, a manufacturing controlling, monitoring and tracking lifecycle activity 250 and an analyzing of manufacturing results lifecycle activity 260. As schematically shown in FIG. 6, selected output from analyzing of manufacturing results lifecycle activity 260 can provide feedback to other lifecycle activities of the lifecycle, such as SW developing and integrating lifecycle activity 210, factory modeling lifecycle activity 230 and manufacturing planning lifecycle activity 240. The output and input interactions between these lifecycle activities will be described in more detail in connection with the description of lifecycle activity 260 for analyzing the manufacturing results.

The various lifecycle activities of FALC 200 of the present invention comprise SW. Hardware parts, equipment or assemblies are needed to support, operate or utilize the SW, which provides the functionality of the six lifecycle activities of FALC 200. The FALC 200 software includes FW SW components. The FW components define common system or factory operation rules and services and they are employed to provide services to application SW components which manage/control processing functions or systems, for example factories including fabricating facilities and various combinations of wafer fab tools, through interaction with the control system of processing equipment, such as on-board wafer fab tool controllers. Application components are adapted to meet the specific requirements of the process and the processing equipment, such as a wafer fab recipe. In other words, repeat processing runs of the same type of product in the same equipment using the same processing conditions does not require a modification in the application components. However, a change in materials, products, equipment or processing conditions typically requires modification of the data of one or more application components. For example, a change in a processing condition requires a corresponding change in one or more application components in order to provide the changed instructions to the equipment. FW components provide the services which enable a user to modify one or more application components to match novel DFS/F to a new processing condition or to a different material or tool. A novel FALC, such as FALC 200, can be adapted to new processing conditions, materials or equipment through data input, provided the adaptation does not require a change to a processing system which is

substantially different. For example, the SW code of several application components of a wafer fab FALC is different from the SW code of the corresponding components of a FALC for a batch manufacturing process of a pharmaceutical product. Application components of the novel DFS/F communicate with equipment such as individual wafer fab tools through protocols and interfaces as will be described more fully in connection with TIC (tool integration component).

FW and application SW elements are referred to as components because they are separate SW entities, each with its own database, server, and standard GUI. The components inter-operate through a public set of communication standards such as DCOM (MICROSOFT® - Microsoft is a registered trademark of Microsoft Corporation, Redmond, Washington- distribute common object model) APIs (application programming interface) or CORBA (common object request broker architecture). SW common building blocks are provided in DFS/F to facilitate the creation of new FW and application components and to modify existing FW and application components. These building blocks typically include GUI (graphical user interface), server and DB (database) elements. Typically, DFS/F and FALC 200 components and SW building blocks are processed by one or more central processing units for data processing or one or more computers. Central processing units and computers which are suitable for the embodiments of the present invention are well known to those of ordinary skill in the art.

The six lifecycle activities of FALC 200, see FIG. 6, include the following functions. SW developing and integrating lifecycle activity 210 is adapted for defining a common structure for factory objects and servers. This common structure simplifies the formation of DFS/F compliant applications. Installing and administrating lifecycle activity 220 installs MES applications. It is necessary to register the capabilities of the MES applications with the DFS/F to make it possible to integrate the MES applications jointly with the capabilities of other applications. Lifecycle activity 220 also monitors and controls the software comprising the factory's MES. Additionally, it regulates access to MES capabilities, providing a common security service. Factory modeling lifecycle activity 230 is adapted for coordinating the formation of a consistent factory model in multiple manufacturing related applications. For example, the introduction of a new product typically requires changes in multiple applications, such as adding the product in

both a WIP (work in progress) application and in a planning application. Lifecycle activity 230 is also adapted for defining the manner in which multiple applications will work together, for example defining how a WIP application and an equipment application can operate together to ensure that the correct equipment is used to manufacture a given product. SW of factory modeling lifecycle activity 230 is adapted for planning, controlling and tracking the manufacturing once a model of the factory's MES functions has been built by using factory modeling related SW.

Manufacturing planning lifecycle activity 240, depicted in FIG. 6, gathers status information from multiple applications in order to provide a planning application component. Additionally, lifecycle activity 240 develops and distributes manufacturing plans/schedules to the application components which manage the factory resources. Manufacturing controlling, monitoring and tracking lifecycle activity 250 is provided to coordinate the functioning of manufacturing application components in the execution of the manufacturing plan/schedule, to produce products according to the steps which are defined in the factory model. Analyzing manufacturing results lifecycle activity 260 is adapted for combining information from application components for analysis. It correlates data in different application components for data analysis and defines, detects and responds to specific factory events. This lifecycle activity is adapted for comparing actual production with planned production and indicating when an updated plan is needed through inputs to SW developing and integrating lifecycle activity 210, factory modeling lifecycle activity 230, and/or manufacturing planning lifecycle activity 240. As illustrated in FIG. 6, feedback loops from lifecycle activity 260 to these lifecycle activities form FALC 200 feedback cycles as follows. Feedback from lifecycle activity 260 to lifecycle activity 210 forms a SW developing cycle 270 while feedback from lifecycle activity 260 to lifecycle activity 280 provides a modeling cycle 280. An executing cycle 290 is formed in the feedback loop from lifecycle activity 260 to lifecycle activity 240. While the six lifecycle activities of novel FALC 200 have been depicted and described in a sequential manner, it will be understood that the SW associated with each of these lifecycle activities typically proceeds concurrently with one or more of the other lifecycle activities.

FALC 200 of the present invention described above is one facet of novel DFS/F. Two other facets of DFS/F are defined as: system layers and N tiers. The system layers aspect of novel DFS/F is described in Table I.

5

Table I
DFS/F System Layers

1. Base technology:

10

This includes the basic technology building blocks for DFS/F such as messaging, graphical user interface (GUI) construction, use of for example MTS (Microsoft transaction manager) for server construction, and mapping objects to a relational database for persistence.

2. Common building blocks:

15

This includes common items such as user-defined attributes, versioning, history and classification schemes. Typical common building blocks are listed and described in Table II.

3. Framework components:

20

FW components manage the overall operation of the factory system through the process of building a factory model, manufacturing products according to the model, and then assessing the outcome to determine the need for improvements. FALC 200 includes these components. Typical FW components are listed and described in Table III.

25

4. Application components:

30

These components comprise factory resource management functionality, such as material management, equipment management and tool integration such as VFEI (virtual factory equipment interface) level communications with tools. Typical application components are listed and described in Table IV.

658220"9963860

The common building blocks of the novel DFS/F are typically utilized to form or modify FW and application components. Representative building blocks are shown in Table II.

Table II
Common Building Blocks

1. Server construction building block which includes interfacing between server API (application programming interface) and factory object instantiation (construction of an object instance).
2. Persistence building block for generating object to SQL/ODBC (structure query language/open DB connection) mapping.
3. DFS/F common GUI controls building block to provide the common controls for the construction of the GUIs.
4. Publish and subscribe messaging building block for publish subscribe messaging, which differs from synchronous DCOM messaging.
5. Dynamic API discovery building block used by FW components to discover services provided by DFS/F components.
6. Associations building block to connect objects across DFS/F components. Representative uses include linking modeling data to answer "where used" questions and linking MES execution information, such as quality data, to equipment history.
7. History building block to provide a common service to store and retrieve the history of factory events.
8. Generic service executor building block to execute the different types of DFS/F services including: (1) synchronous services, (2) long running services

which are implemented through a completion callback protocol and (3) GUI based services that are implemented through a link between other services and a target computer desktop.

- 5 9. Classifications building block which provides a common service to classify factory objects for queries and analysis.
- 10 10. Customer defined attributes building block for extending the DFS/F component based object models. DFS/F typically provides user extensible attribute models which support expansion of the model based on the user's needs.
- 15 11. State models building block includes a service to define and then operate state models, such as tracking states for equipment, material and ECNs (engineering change notice). A state model building block can be defined such that it tracks unplanned downtime of the various chambers of a cluster tool.
- 20 12. Namespace building block defines management areas within DFS/F data models.
- 25 13. Schedule/datebook building block to support scheduling and add the capability for each factory object to have a datebook showing future events and history enhancements to indicate how a particular schedule was followed.
- 30 14. Templates building block provides services to define base definitions that can be shared among factory objects. For example, this building block can be used to define common characteristics for all memory products.
15. Versioned objects building block includes services to manage and track changes in factory model objects over time, e.g. for example managing and tracking changes in the definition of a particular product over time.

05363966-072999

16. Navigation building block to view the relationship between factory objects such as a material lot history.

FW components utilized in novel FALC 200 define common rules and services which are utilized by application components, see Table IV, through steps in the FALC 200. Examples of suitable FW components are provided in table III.

Table III
Framework Components

1. SC (security component) provides basic security having 3 security modes: (1) defining user roles, (2) assigning users to roles and (3) defining access to DFS/F objects and methods by role.
2. GCC (GUI console component) is a container supporting navigation and showing of data between DFS/F GUIs. Custom and/or third party applications can be added to the console toolbar. A custom environment can be created by adding GUIs and factory objects. The console can include a transaction view, i.e. full screen, cascade, tile and icons. A GUI console navigator allows browsing and selecting DFS/F factory objects and has search capability based on server-supplied search criteria. For example, it provides navigation and data sharing between ActiveX GUIs.
3. PLMC (performance & license management component) for tracking and regulating system usage of components.
4. SMC (saga management component) provides support for “long running” transactions that should be treated as a unit but take too long to rely on standard DB locking techniques.
5. CRC (context resolution component) aids in MES execution by linking context to results, insuring that the appropriate instructions are delivered to any

resource by allowing users to flexibly model how resource selections are made.

- 5 6. CMC (configuration management component) provides the management of factory model changes across components.
7. CC (calendar component) provides calendar and shift definitions for scheduling and reporting.
- 10 8. VWC (visual workflow component) defines and executes manufacturing processes and is capable of executing predetermined business processes. VWC defines business processes graphically as a sequence/network of service invocations from a palette of DFS/F services. Other DFS/F components utilize the VWC for process definitions. For example, the WIP management
15 component uses VWC services to define how products are produced and uses it to execute the processing of material lots. VWC process definition capability includes the exchange of data between service invocations and control structures to determine/select the path(s) through predefined business processes. VWC is adapted for executing business processes autonomously, i.e. functioning independently of other SW components, and is capable of
20 responding to automated inputs as well as to user inputs.
9. RCC (resource coordination component) is responsible for having active resources available at dispatch stations. Matches resources to common
25 jobs/batches employing BRC. Together with BRC it coordinates rendezvous of active and passive resources.
10. EVMC (event monitor component) monitors/subscribes to events published by DFS/F services. A DFS/F service can be executed (including launching a
30 VWC job) when a monitored event occurs. EVMC supports vigilant manufacturing through the creation of factory monitors.

11. BRC (bill of resources component) establishes the resources across multiple DFS/F components needed to launch a batch process, i.e. a batch process involving the coordinated action of multiple resources.

12. DMC (data manager component) consolidates data from FW components and application components for reporting and analysis. It is based on DW (data warehouse) technology and can provide sample DW star schema and reports. DMC can access DBs for unstructured data analysis.

Application components provide the MES instructions to manufacturing equipment to manage and control specific tools and processes. Examples of suitable application components are described in Table IV.

Table IV
Application Components

1. QMC (quality management component) provides quality analysis and flexible data collection. It is able to determine corrective manufacturing tactics in order to ensure conformance to predetermined business rules.

2. TIC (tool integration component) providing two way communications between DFS/F and diverse equipment types. It is adapted for communicating through tool protocols such as SECS (SEMI – Semiconductor Equipment and Materials International – Equipment Communication Standard), GEM (generic equipment model) and VFEI (virtual factory equipment interface). SECS, GEM and VFEI are tool protocols which are well known to those of ordinary skill in the art.

3. EMC (equipment management component) resolves the different equipment states in SEMI E10 states, using a novel hierarchical model to track individual tools in chambers in cluster tools.

4. RMC (recipe management component) providing definition, selection and distribution of recipes to equipment such as fab tools.

5. DSC (dispatching and scheduling component) for scheduling and dispatching of factory tasks including processing and maintenance.

5 6. MHC (material handling component) for interfacing with materials handling equipment.

10 7. WMC (WIP –work in progress- management component) is provided to support tracking of wafers, lots, batches and carrier, supporting proactive business decisions for example: “if, then”. WIP offers cluster tool visibility and control thereby providing MES to wafer lots inside the cluster tool.

15 8. Legacy system interface which is a component to access existing factory software.

20 A third facet of DFS/F of the present invention comprises the various tiers which can be present in the FW components, application components and common building blocks. For example, this can be a 3 tier facet as follows. A first tier includes clients using services from other SW programs or components, e.g. visual WF jobs, GUIs and customer programs, such as VB (virtual Basic). A second tier comprises application or FW servers using for example MTS/DCOM to communicate DCOM. A third tier includes a DB engine, such as Oracle using an ODB (open database connectivity) interface. All DFS/F components employ these three tiers, while common building blocks can be used in one or more of these tiers depending on the structure and function of the building block.

25 FW components are associated with various lifecycle activities of FALC 200. Representative examples of these associations are shown in Table IV.

Table V

30 FW Components Associated With FALC 200 lifecycle activities

FALC 200 lifecycle activity	FW Component
--------------------------------	--------------

220	SC, GCC, PLMC, SMC
230	CRC, CMC, CC
250	VWC, RCC, EVMC, BRC
5 260	DMC

FW components of an FALC of the present invention, such as FALC 200, are adapted for interacting with each other, thereby working together. For example, a factory model can be revised as illustrated in FIG. 7, by utilizing an exchange of messages between a GCC (GUI console component) 310, a CMC (configuration management component) 312 and a VWC (visual workflow component) 314. Changes in the factory model are collected to form an ECN (engineering change notice) in a CMC such as CMC 312 depicted in FIG. 7. The ECN is opened 316 (FIG. 7) to collect the changes in the factory model. The resulting ECN data 318 are employed by GCC 310 during the editing of the VWC workflow 320. The workflow is displayed and edited through a GUI 322 contained within GCC 310. The changed VWC workflow is returned to VWC in step 324, and VWC 314 then adds the changed workflow to the CMC ECN in step 326.

Any application component of the novel DFS/F can participate in the DFS/F by implementing the needed services from the appropriate FW components, thereby forming a plug and play type of SW framework, as is schematically illustrated in FIG. 8, employing FW or application servers working together through exchanges of messages. As shown in FIG. 8, FW components using FW component servers for CMC 410, VWC 412, DMC 414, EMC 416, GCC 418 and SC 419 use common service protocols to provide services to application components using application component servers for EMC 420, WMC 422, DSC 424, QMC 426, TIC 428 and gateway component 429. These services communicate through common service protocols 430 using for example DCOM communications. This SW technique of the present invention enables a user to modify processing and equipment MES instructions through data inputs rather than coding thus eliminating the need for time consuming programming changes requiring specialized skills. The application components require effective communications with integrated pieces of equipment such as wafer fab tools and materials handling equipment in order to execute the MES instructions to the controllers of wafer fab tools and/or other equipment.

Typical controllers include processors for example micro processors such as on-board computers, computer operated software and mechanical/electrical controllers such as switches and electrical circuits employing for example a variable resistor such as a potentiometer.

5

DFS/F of the present invention includes a novel TIC application component (Table IV) for facilitating communications between various DFS/F components and equipment, such as wafer fab tools, by providing the basic equipment control building blocks which can be assembled in a VWC (Table III) workflow to control a machine. TIC provides services which include sending and receiving VFEI messages or commands to and from equipment. A sequence of these commands or messages represents a business logic such as commands to control a tool. These types of sequences can be defined in a VWC workflow. TIC provides the building blocks which are used to send commands or messages to a tool and to receive messages or information from the tool, and to communicate the received messages or information to other DFS/F components. TIC is part of the communications link between DFS/F and equipment such as wafer fab tools.

10

15

20

25

TIC of the present invention comprises a novel combination of a TIP (tool interface program) and a novel TIC adapter. TIP is provided for each machine or tool type to translate VFEI commands or messages to an interface, such as SECS, of a machine and its controls such as control SW. Equipment which is adapted for communicating with novel DFS/F will have a TIP instance, i.e. a SW process dedicated to the equipment, running as an intermediary between the equipment and the DFS/F. An example of a suitable communication protocol between a TIP instance for a machine and DFS/F is a VFEI on DCOM protocol. Additionally it is contemplated to provide a novel VFEI+ on DCOM protocol wherein VFEI+ will include enhancements for administration, for modeling and for the RPC (remote procedure call) nature of DCOM. It is contemplated to distribute TIP on several computers, for example where these computers are utilized in computer integrated wafer fab tools employing TIP SW.

30

A TIC adapter of the present invention is an intermediary between DFS/F and TIPs for example by allowing other DFS/F components to access tool capabilities through generalized commands which the adapter then adapts to the needs of the tool's specific

TIP instances. This is illustrated in the following example for collecting measurements using a metrology tool in a wafer fab having different tools available for collecting the measurements. These novel techniques are adapted for embedding the collection process in a VWC WF which then enables a user, such as a process technician or engineer, to send identical measurement requests to TIC. The TIC adapter then translates the request into tool specific VFEI requests based on the particular tool which is selected at run time. These techniques of the present invention are capable of defining for example a single QMC (quality management component) data collection plan which can then be used for different tools collecting the same types of data, because the TIC adapter can translate the data parameter names from the generic name in the QMC plan to the specific names required by specific equipment types or tools.

Advantageously, a TIC adapter can also perform a protocol conversion between DFS/F style communications and the TIPs. This is a useful function because many DFS/F components are constructed using MTS and are designed to be transactional and stateless. For example, DFS/F servers both request and process services are either “synchronous”, i.e. services that are expected to be completed within a predictably short completion time of one second or less, and services that are considered long running and that use an LRSP (long running service protocol). As defined herein, the term “long running service” includes services having a completion time which cannot be predicted and which are thus unsuitable for DB locking techniques which are typically employed in connection with conventional services, such as synchronous services. It is contemplated to process the service by an MTS based server using a relatively short lived MTS thread and wherein any longer term state is saved in a DB. It is also contemplated that service requests can be based on a DCOM RPC style model, where a service is requested which is then followed by a returned reply. The TIC adapter is an intermediary between DFS/F and TIPs.

An example of a novel TIC is illustrated in FIG. 9, showing a novel DFS/F including TIC 520 of the present invention. In the present example, a distributed computer having nodes A, B and C was used. DFS/F components 510 communicated with fab tools 538 and 548 by means of TIC 520. MES instructions for tools 538 and 548 were sent by components 510 to novel TIC adapter 522 of TIC 520. For example, instructions for starting the processing of a material can be communicated from a DFS/F component 510,

such as a VWC workflow, to tool 538 through TIC 520. The TIC adapter was accessed through an MTS server on node A. The instructions for tool 538 were communicated from TIC adapter 522, through a VFEI protocol on DCOM 532, to a TIP instance 534 on node B. TIP instance 534 then communicated these instructions to tool 538 using SECS protocol 536 of tool 538. Similarly, the instructions for tool 548 were communicated from TIC adapter 522 to a TIP instance 544 on node C, through a VFEI protocol on DCOM 542. TIP instance 544 communicated the instructions to tool 548 using SECS protocol 546 of tool 548. Also a DB (not shown) can be provided which can be accessed by TIC adapter 522 to store and retrieve TIC data such as the network locations of specific TIPs, using such methods and techniques as are well known to those of ordinary skill in the art.

TICs of the present invention employ Microsoft DCOM messaging technology to send messages between DFS/F components of the present invention and a TIC adapter, and between a TIC adapter and tool interface program instances according to the present invention. A TIC adapter is constructed using common building blocks of the present invention such as those described in Table II. An inventive TIC adapter can be constructed using a Microsoft transaction server. The adapter is capable of simultaneously managing messages from many DFS/F components and many TIP instances. A TIC adapter of the present invention is capable of saving information such as pending event requests, i.e. the instruction to a TIP regarding a request that a tool should report a particular status or operating event. The information can for example be saved using Oracle database technology using such techniques as are well known to those of ordinary skill in the art.

FIG. 10 shows a sequence of messages from a VWC 610 executing a business process which is defined in a VWC WF, to a TIC adapter 612 and then on to a TIP instance 614. This TIP instance includes a tool interface program for interfacing with processing equipment (not shown) such as a wafer fab tool for manufacturing an integrated circuit structure. The VWC business process sends an instruction to TIC adapter 612, requesting an event setup 616 which is a request for reporting an equipment event, for example reporting the completion of a wafer fabrication process. The event setup request is then routed 618 to TIP instance 614, which is subsequently acknowledged 620 to VWC 610. VWC then requests an event report 622 concerning the event setup

request from TIC adapter 612. Event report request 622 is acknowledged in step 624. When the requested event occurs on the equipment, a report concerning the event is communicated 626 from TIP instance 614 to TIC adapter 612, which then checks to determine if there is a pending report request. When there is a pending report request, such as request 622, TIC adapter 612 delivers the requested report 628 to the VWC WF and acknowledges delivery 630 of the event report to TIP instance 614. Optionally, the VWC business WF can execute a loop requesting additional event reports. Once the requested reports are received from the tool, the loop terminates and VWC WF will send a concluding command to TIC to notify the tool, via its TIP, that the requested event reports should no longer be sent. It will be noted that TIP instance 614 might need a protocol such as SECS (not shown) in order to communicate with the equipment.

As shown in the above examples, application components such as TIC, communicate with and utilize FW components such as VWC (Table III). Additionally, a particular FW component can communicate with another FW component and utilize its services. For example, all FW components can use the common security FW component to regulate access to the services which these FW components provide, as is illustrated in FIG. 8.

Plug and play capabilities of a VWC are illustrated for example by WFs wherein the VWC is a technique for defining a business process as a WF which is subsequently executed as a job. WFs are graphical representations of business processes defined for manufacturing a product such as a semiconductor structure typically including many processing steps, equipment steps, movement steps, decision steps and data steps. Each of these steps is defined as a step in a WF. WFs can include movements, decisions, quality activities and equipment commands. WFs are built using drag and drop techniques in a computer graphical environment using techniques which are well known to those of ordinary skill in the art. Once defined, a WF is executed as a job. For example, a WF can be executed to create a lot of a product.

DFS/F of the present invention can be utilized to start production, such as a wafer fab run, by means of the WIP management application component (Table IV) as follows. A user, such as a process engineer, defines how a particular product is made by creating a

WF in VWC that defines the sequence of steps needed to make the product. This sequence can for example include a wafer fab recipe. The user then starts the lot by using the WIP management application component service or GUI requesting VWC to start the WF for the lot. In summary, WIP uses VMC as a service for defining the processing steps and subsequently for actually manufacturing/fabricating production lots.

Building blocks provide common functionality to the novel DFS/F techniques to facilitate the construction of application and FW components. For example, a DFS/F classification building block (Table II) can be used in the EMC application component (Table IV) to classify equipment. It can also be used in the WIP management application component (Table IV) to classify products. This building block includes the three DFS/F tiers. A first tier uses a DFS/F GUI within the GCC to view and modify or update classification information. A second tier includes SW code/logic that is provided in the component server, such as EMC or WMC servers. This particular SW defines how the classifications are specified. For example, this SW enables a user to classify a particular wafer fab tool as an etcher. The classification building block SW is used in the component servers through for example a C++ inheritance capability, i.e. the capability to pass resources or attributes from component servers down to the specifics of a particular product, process or machine. A third tier comprises a DB table definition that is employed by the component server logic for storing and retrieving classification information.

Digitally coded data structures or information of novel DFS/F or its components and building blocks can be stored on a removable electronic data storage medium or device, such as computer floppy disks, removable computer hard disks, magnetic tapes and optical disks, to facilitate the use of the same technique at different processing locations. Alternatively, the data structure or information can be stored on a non-removable electronic data storage medium, including a medium positioned at a location which is remote from the tool, using such storage devices as are well known to those of ordinary skill in the art. The data structures or information can be communicated from a remote location to a central data processing unit or to a computer using communications techniques which are well known to those of ordinary skill in the art including hard wire connections, wireless connections and data communications methods utilizing one or more modems or techniques using one or more computers commonly known as servers.

Once DSF/F or its components or building blocks are installed, they are commonly stored in a data storage device or memory of a dedicated computer or a distributed computer system which is integrated with the processing system and its equipment, using such data storage techniques as are well known to those of ordinary skill in the art.

5

It is contemplated to provide novel products comprising digitally coded data structures or information of novel DFS/F or its components or building blocks, stored in memory such as a removable electronic data storage medium or device. It is also contemplated to provide a novel computer integrated apparatus having a memory or data storage device for storage of electronic or digital data, a central processing unit or a computer, and processing equipment wherein DFS/F or its components or building blocks are stored in the data storage device.

The invention has been described in terms of the preferred embodiment. One skilled in the art will recognize that it would be possible to construct the elements of the present invention from a variety of means and to modify the placement of components in a variety of ways. While the embodiments of the invention have been described in detail and shown in the accompanying drawings, it will be evident that various further modifications are possible without departing from the scope of the invention as set forth in the following claims.

036366-02899

We claim:

- 5 1. A method for defining a computer implemented factory automation lifecycle,
the method comprising:
 - a) Defining, installing and administrating lifecycle activity framework
components;
 - b) defining factory modeling lifecycle activity framework components; and
 - 10 c) defining manufacturing controlling, monitoring and tracking lifecycle
activity framework components.
- 15 2. The method of claim 1, wherein defining administrating lifecycle activity
framework components comprises defining one or more framework
components selected from the group consisting of a security component, a GUI
console component, a performance and license management component and a
saga management component.
- 20 3. The method of claim 1, wherein defining factory modeling lifecycle activity
framework components comprises defining one or more framework
components selected from the group consisting of a context resolution
component, a configuration management component and a calendar
component.
- 25 4. The method of claim 1, wherein defining manufacturing controlling,
monitoring and tracking lifecycle activity framework components comprises
defining one or more framework components selected from the group
consisting of a visual workflow component, a resource coordination
component, an event monitor component and a bill of resources component.
- 30 5. The method of claim 1 additionally comprising a method for defining one or
more analyzing of manufacturing results lifecycle activity framework
components.

6. The method of claim 5, wherein defining one or more analyzing of manufacturing results lifecycle activity framework components comprises defining a data manager component.
7. The method of claim 5 additionally defining a method for defining interactions between the one or more manufacturing results lifecycle activity framework components and components selected from the group consisting of factory modeling lifecycle activity framework components.
8. The method of claim 1 additionally defining a SW developing and integrating lifecycle activity.
9. The method of claim 1 additionally defining a manufacturing planning lifecycle activity.
10. The method of claim 1 wherein defining a factory automation lifecycle comprises defining a factory automation lifecycle for processing an integrated circuit structure.
11. The method of claim 1, wherein defining a factory automation life-cycle additionally comprises defining framework components such that the framework components are adapted for communicating with a tool integration component, wherein the framework components are selected from the group consisting of installing and administrating lifecycle activity framework components, factory modeling lifecycle activity framework components, and manufacturing controlling, monitoring and tracking lifecycle activity framework components.
12. The method of claim 11 wherein defining manufacturing controlling, monitoring and tracking lifecycle activity components comprises defining a visual workflow component.
13. A method for managing a processing system including one or more computers,

the method comprising:

- a) running a factory automation lifecycle including one or more framework software components by means of the one or more computers;
- b) running one or more application software components to provide one or more computer implemented instructions for managing the system, wherein the one or more framework components are adapted for managing the application components;
- c) determining whether the one or more instructions need to be modified;
- d) communicating the one or more instructions to the system, if the instructions do not need to be modified;
- e) modifying the instructions if they need to be modified, by means of the one or more framework components thereby forming modified instructions; and
- f) communicating the modified instructions to the system.

14. The method of claim 13 additionally comprising managing the processing system by executing the one or more instructions in the system.

15. The method of claim 13 wherein running one or more of the framework components comprises running one or more components selected from the group consisting of a security component, a GUI console component, a performance and license management component, a saga management component, a context resolution component, a configuration management component, a calendar component, a visual workflow component, a resource coordination component, an event monitor component, a bill of resources component and a data manager component.

16. The method of claim 13 wherein running one or more application components comprises running one or more components selected from the group consisting of a quality management component, a tool integration component, an equipment management component, a recipe management component, a dispatching and scheduling component, a material handling component, a work in progress component and a legacy system interface component.

17. The method of claim 13 wherein communicating comprises communicating by means of a tool integration component.

5 18. The method of claim 17 wherein communicating comprises communicating by means of:

- a) a tool interface program; and
- b) a tool integration component adapter.

10 19. The method of claim 17 wherein the one or more computer implemented instructions are visual framework component instructions.

15 20. The method of claim 13, additionally comprising forming one or more framework components by means of one or more software building blocks selected from the group consisting of a server construction building block, a persistence building block, a common GUI controls building block, a publish and subscribe messaging building block, a dynamic API discovery building block, an associations building block, a history building block, a generic service executor building block, a classifications building block, a customer defined attributes building block, a state models building block, a namespace building block, a schedule/datebook building block, a templates building block, a versioned objects building block and a navigation building block.

20 21. The method of claim 13 wherein running one or more framework components additionally comprises communicating a data structure of the one or more framework component to a data structure of the one or more components selected from the group consisting of framework components and application components.

25 22. The method of claim 13, wherein modifying the instructions comprises inputting data.

30 23. The method of claim 13 wherein managing a processing system comprises

666220"99663660

managing a system for processing an integrated circuit structure.

24. The method of claim 23 wherein the system comprises one or more wafer fabrication tools.

25. A method for linking a software framework to an apparatus, the method comprising linking by means of a tool integration component including:

- a) a tool interface program; and
- b) a tool integration component adapter.

26. The method of claim 25 wherein linking comprises exchanging messages between:

- a) the one or more framework components;
- b) the tool integration component adapter; and
- c) the tool interface program.

27. The method of claim 26 wherein the one or more framework components comprise a visual workflow component.

28. A method for processing a product, the method comprising:

- a) determining specifications for processing the product; and
- b) managing the processing by means of a distributed factory system framework including: (1) a factory automated lifecycle having one or more framework components and (2) one or more application components wherein the framework components are adapted for managing the application components.

29. The method of claim 28 wherein managing additionally comprises:

- a) determining whether the distributed factory system framework needs to be modified in order to meet the specifications; and
- b) modifying one or more of the application components if the distributed factory system framework needs to be modified.

30. The method of claim 29 wherein modifying comprises inputting data.

31. The method of claim 28 wherein managing additionally comprises forming one or more framework components by means of one or more software building blocks.

32. The method of claim 28 wherein managing additionally comprises:

- a) forming one or more computer implemented instructions for managing, by means of the one or more application components;
- b) communicating the one or more instructions to equipment for processing the product; and
- c) executing the one or more instructions on the equipment.

33. The method of claim 32 wherein communicating comprises communicating by means of a tool integration component, wherein the tool integration component comprises: (1) a tool integration component adapter and (2) a tool interface program.

34. The method of claim 28 wherein processing a product comprises processing an integrated circuit structure.

35. A method for starting a wafer fab run, the method comprising:

- a) determining a sequence of processing steps for processing the wafer fab run;
- b) forming a workflow defining the sequence, in a visual workflow component which is included in a distributed factory system framework having: (1) framework components and (2) application components; and
- c) requesting the visual workflow component to start the run by means of a service which is a work in progress management component or a GUI within a GUI console component.

36. An apparatus for processing a product, the apparatus comprising:

- a) product processing equipment;

- b) at least one central processing unit for electronic data processing;
- c) a link for operably linking the central processing unit to the product processing equipment;
- d) a memory for storing digitally coded data structures, wherein the memory is operably linked to the at least one central processing unit; and
- e) a digitally coded first data structure stored in the memory wherein the data structure comprises a factory automation lifecycle including: (1) administrating lifecycle activity framework components, (2) factory modeling lifecycle activity framework components, and (3) manufacturing controlling and tracking lifecycle activity framework components.

37. The apparatus of claim 36, wherein the administrating lifecycle activity framework components comprise one or more framework components selected from the group consisting of a security component, a GUI console component, a performance and license management component and a saga management component.

38. The apparatus of claim 36, wherein the factory modeling lifecycle activity framework components comprise one or more framework components selected from the group consisting of a context resolution component, a configuration management component and a calendar component.

39. The apparatus of claim 36, wherein the manufacturing controlling and tracking lifecycle activity framework components comprise one or more framework components selected from the group consisting of a visual workflow component, a resource coordination component, an event monitor component and a bill of resources component.

40. The apparatus of claim 36 additionally comprising one or more analyzing of manufacturing results lifecycle activity framework components.

41. The apparatus of claim 40, wherein the one or more analyzing of manufacturing results lifecycle activity framework components comprise a data

manager component.

42. The apparatus of claim 36 additionally comprising a digitally coded second data structure including application components, wherein the first data structure is adapted for managing the second data structure.

43. The apparatus of claim 42 additionally comprising a digitally coded third data structure including software building blocks for forming one or more of the framework components.

44. The apparatus of claim 36, wherein the first data structure comprises:

- a) a digitally coded fourth data structure including a GUI console component; and
- b) a digitally coded fifth data structure including a configuration management component.

45. The apparatus of claim 44, wherein the fourth and fifth data structures are adapted for interacting.

46. The apparatus of claim 36 wherein the link comprises a tool integration component including: (1) a tool integration component adapter and (2) a tool interface program

47. The apparatus of claim 36 comprising an apparatus for processing an integrated circuit structure.

48. An apparatus for processing a product, the apparatus comprising:

- a) product processing equipment;
- b) at least one central processing unit for electronic data processing;
- c) a link for operably linking the central processing unit to the product processing equipment;
- d) a memory for storing digitally coded data structures, wherein the memory is operably linked to the at least one central processing unit; and

- e) a distributed factory system framework for managing the product processing, the distributed factory system framework comprising: (1) a digitally coded first data structure comprising a factory automation lifecycle including digitally coded framework components, (2) a digitally coded second data structure comprising application components adapted for communicating digitally coded instructions to the processing equipment, wherein the first data structure is adapted for managing the second data structure and (3) a link for communicating the digitally coded instructions to the processing equipment.

49. The apparatus of claim 48 comprising an apparatus for processing an integrated circuit structure.

50. A distributed factory system framework for managing a processing system, the distributed factory system framework comprising:

- a) a digitally coded first data structure comprising digitally coded framework components;
- b) a digitally coded second data structure comprising application components adapted for communicating digitally coded instructions to the processing system, wherein the first data structure is adapted for managing the second data structure; and
- c) a link for communicating the digitally coded instructions to the processing system.

51. The distributed factory system framework of claim 50 wherein the framework components comprise one or more components selected from the group consisting of a security component, a GUI console component, a performance and license management component, a saga management component, a context resolution component, a configuration management component, a calendar component, a visual workflow component, a resource coordination component, an event monitor component, a bill of resources component and a data manager component.

52. The distributed factory system framework of claim 50 wherein the application components comprise one or more components selected from the group consisting

of a quality management component, a tool integration component, an equipment management component, a recipe management component, a dispatching and scheduling component, a material handling component, a work in progress component and a legacy system interface component.

5

53. The distributed factory system framework of claim 50 wherein the link comprises a fourth data structure including a tool integration component.

54. The distributed factory system framework of claim 53 wherein the fourth data structure comprises:

10

- a) a tool interface program fifth data structure; and
- b) a tool integration component adapter sixth data structure.

55. The distributed factory system framework of claim 50, additionally comprising one or more software building blocks selected from the group consisting of a server construction building block, a persistence building block, a common GUI controls building block, a publish and subscribe messaging building block, a dynamic API discovery building block, an associations building block, a history building block, a generic service executor building block, a classifications building block, a customer defined attributes building block, a state models building block, a namespace building block, a schedule/datebook building block, a templates building block, a versioned objects building block and a navigation building block.

15

20

56. A data storage device comprising a digitally coded first data structure including a factory automation lifecycle having:

25

- a) administrating lifecycle activity framework components;
- b) factory modeling lifecycle activity framework components; and
- c) manufacturing controlling and tracking lifecycle activity framework components.

30

57. The device of claim 56, wherein the administrating lifecycle activity framework components comprise one or more framework components selected from the group consisting of a security component, a GUI console component, a

performance and license management component and a saga management component.

5 58. The device of claim 56, wherein the factory modeling lifecycle activity framework components comprise one or more framework components selected from the group consisting of a context resolution component, a configuration management component and a calendar component.

10 59. The device of claim 56, wherein the manufacturing controlling and tracking lifecycle activity framework components comprise one or more framework components selected from the group consisting of a visual workflow component, a resource coordination component, an event monitor component and a bill of resources component.

15 60. The device of claim 56 additionally comprising one or more analyzing of manufacturing results lifecycle activity framework components.

20 61. The device of claim 60, wherein the one or more analyzing of manufacturing results lifecycle activity framework components comprise a data manager component.

62. The device of claim 56 comprising a plurality of framework components which are adapted for interacting with a GUI console framework component.

25 63. The device of claim 56 additionally comprising a digitally coded second data structure including application components, wherein the first data structure is adapted for managing the second data structure.

30 64. The device of claim 63 additionally comprising a digitally coded third data structure including software building blocks for forming one or more of the framework components.

65. The device of claim 64 wherein the first, second and third data structures are

adapted for processing an integrated circuit structure.

66. A data storage device comprising:

- a) a digitally coded first data structure comprising a factory automation lifecycle including digitally coded framework components; and
- b) a digitally coded second data structure comprising application components, wherein the first data structure is adapted for modifying the second data structure.

67. The device of claim 66 wherein the framework components comprise one or more components selected from the group consisting of a security component, a GUI console component, a performance and license management component, a saga management component, a context resolution component, a configuration management component, a calendar component, a visual workflow component, a resource coordination component, an event monitor component, a bill of resources component and a data manager component.

68. The device of claim 66 wherein the application components comprise one or more components selected from the group consisting of a quality management component, a tool integration component, an equipment management component, a recipe management component, a dispatching and scheduling component, a material handling component, a work in progress component and a legacy system interface component.

69. The device of claim 66 additionally comprising a digitally coded third data structure including one or more software building blocks selected from the group consisting of a server construction building block, a persistence building block, a common GUI controls building block, a publish and subscribe messaging building block, a dynamic API discovery building block, an associations building block, a history building block, a generic service executor building block, a classifications building block, a customer defined attributes building block, a state models building block, a namespace building block, a schedule/datebook building block, a templates building block, a versioned objects building block and a navigation

The present invention provides a novel distributed factory system framework including a novel factory automation lifecycle (200) having lifecycle activities for SW developing and integrating (210), installing and administrating (220), factory modeling (230), manufacturing planning (240), manufacturing controlling, monitoring and tracking (250) and analyzing of manufacturing results (260). The factory lifecycle comprises framework components. The distributed factory system framework also includes application components and SW building blocks. The framework components are adapted for managing the application components, while the application components are utilized to provide instructions for managing a process such as a wafer fab. The building blocks are adapted for forming or modifying framework and application components. The distributed factory system framework provides computer implemented methods for integrating processing systems and facilitates process and equipment changes.

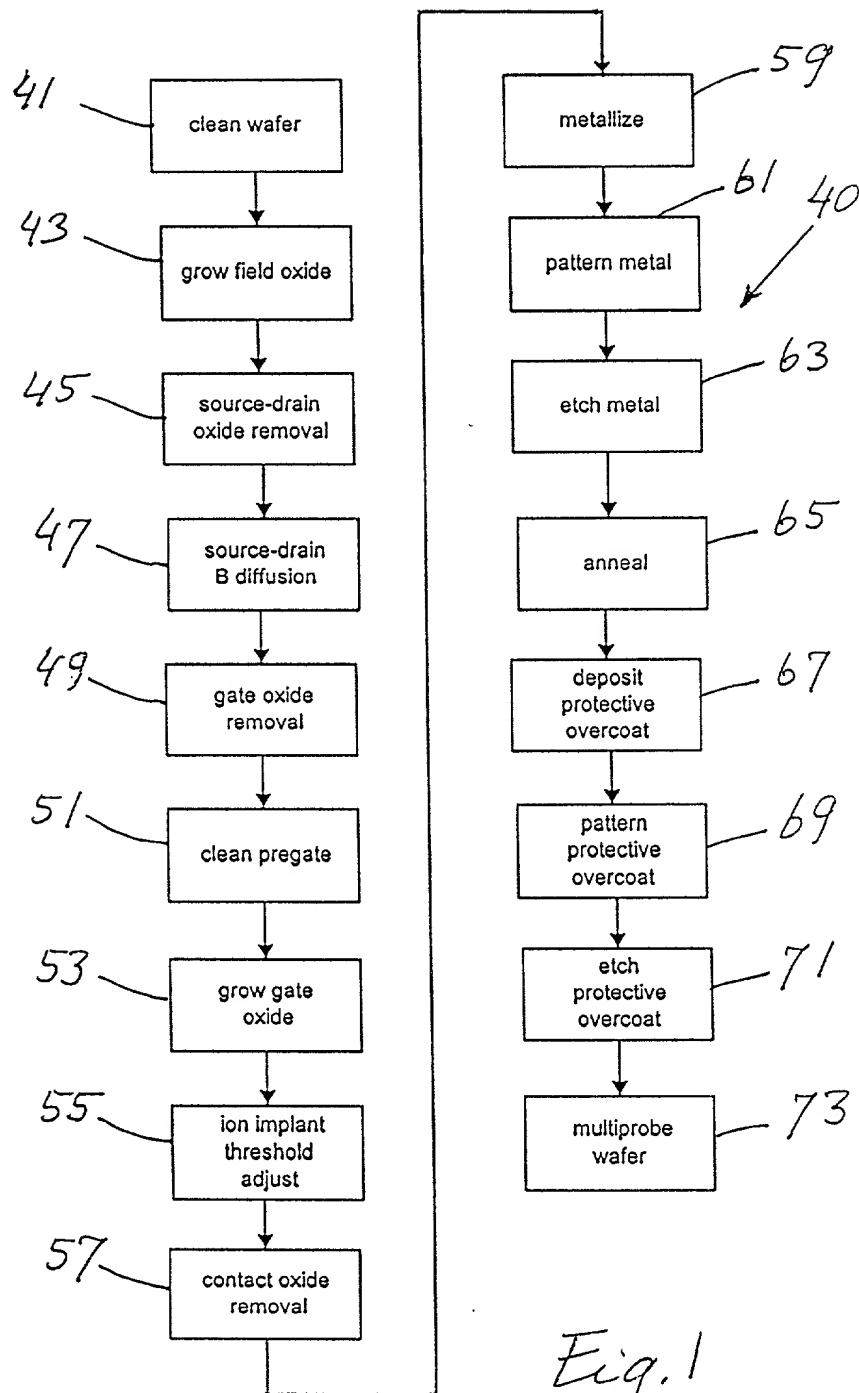


Fig. 1
Prior Art

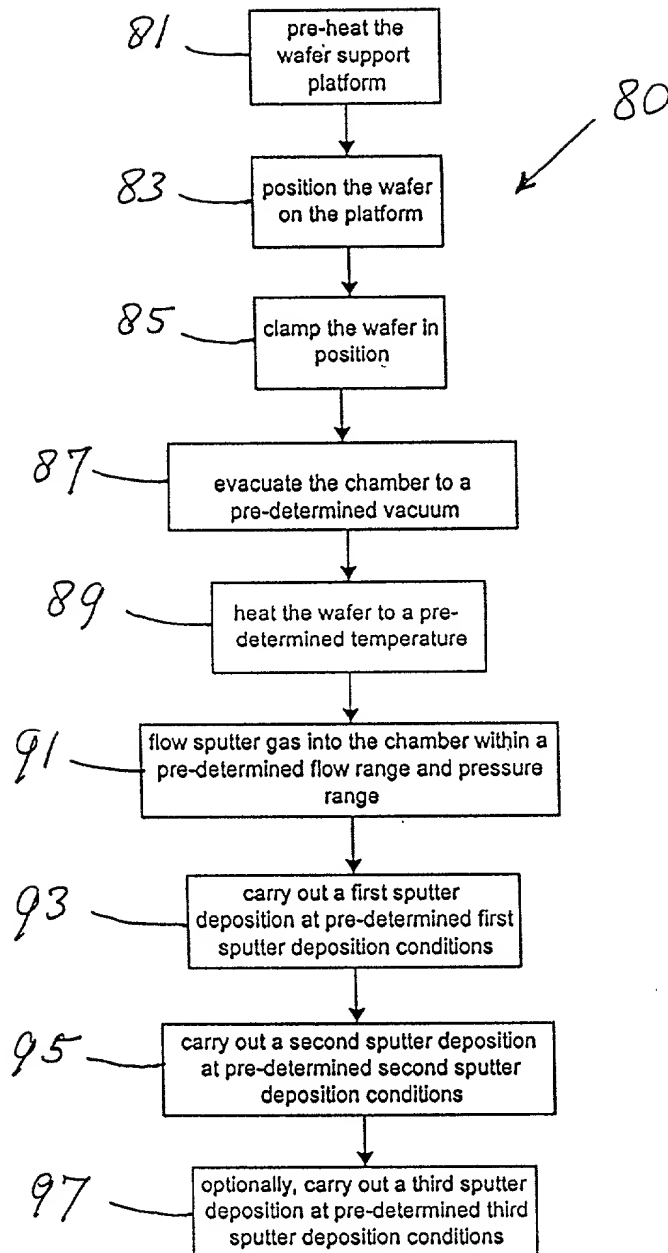


Fig. 2
Prior Art

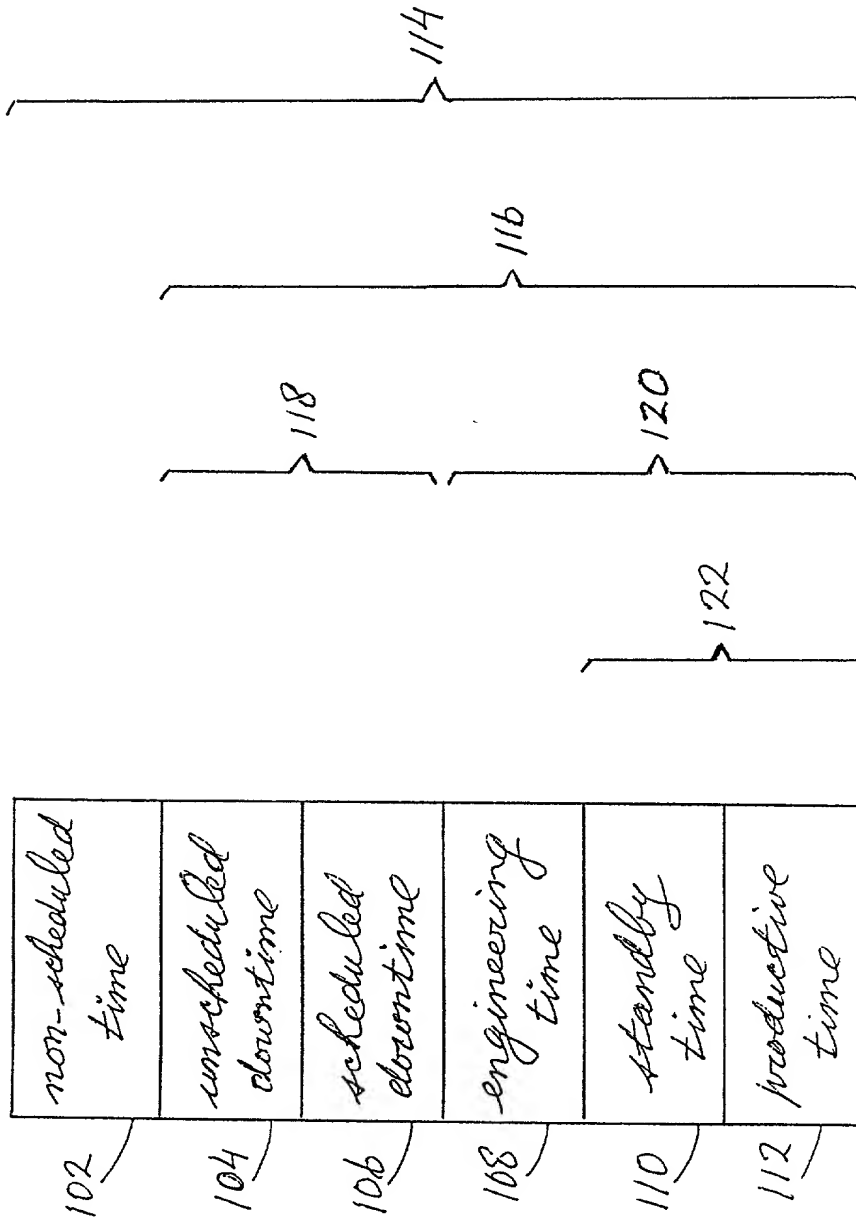


Fig. 3
Prior Art

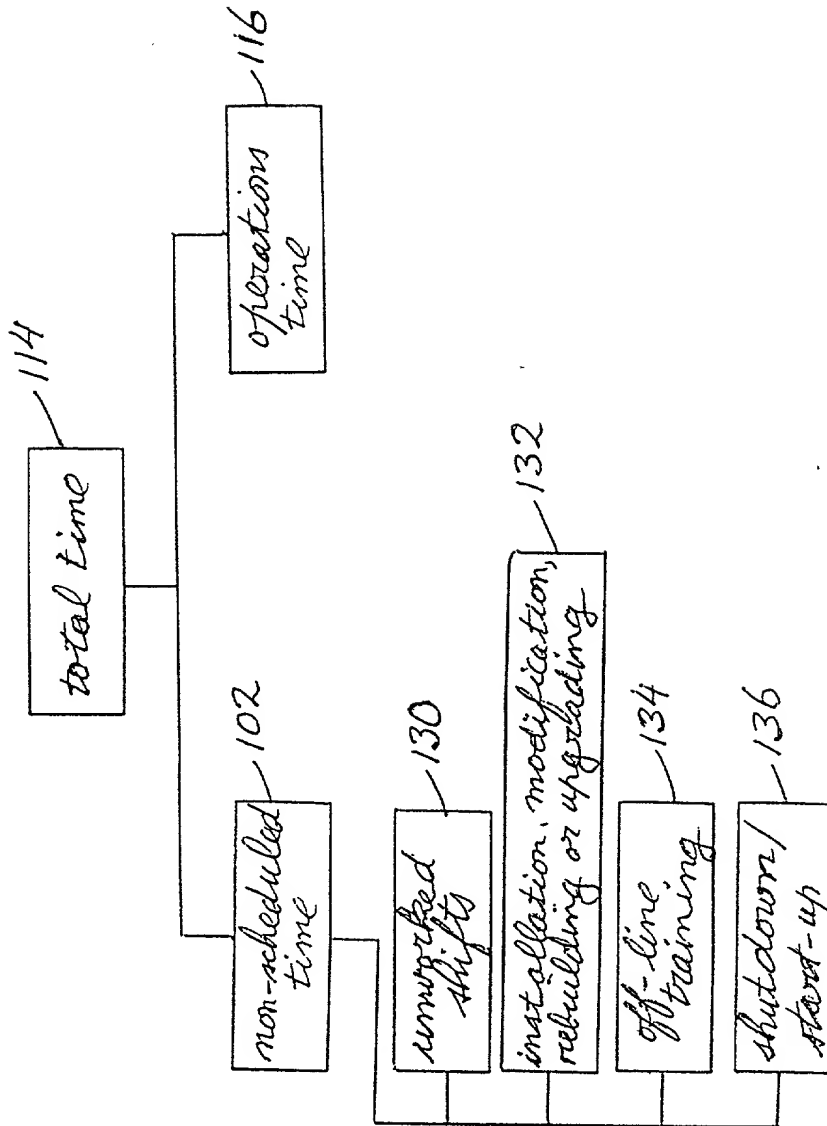


Fig. 4
Prior Art

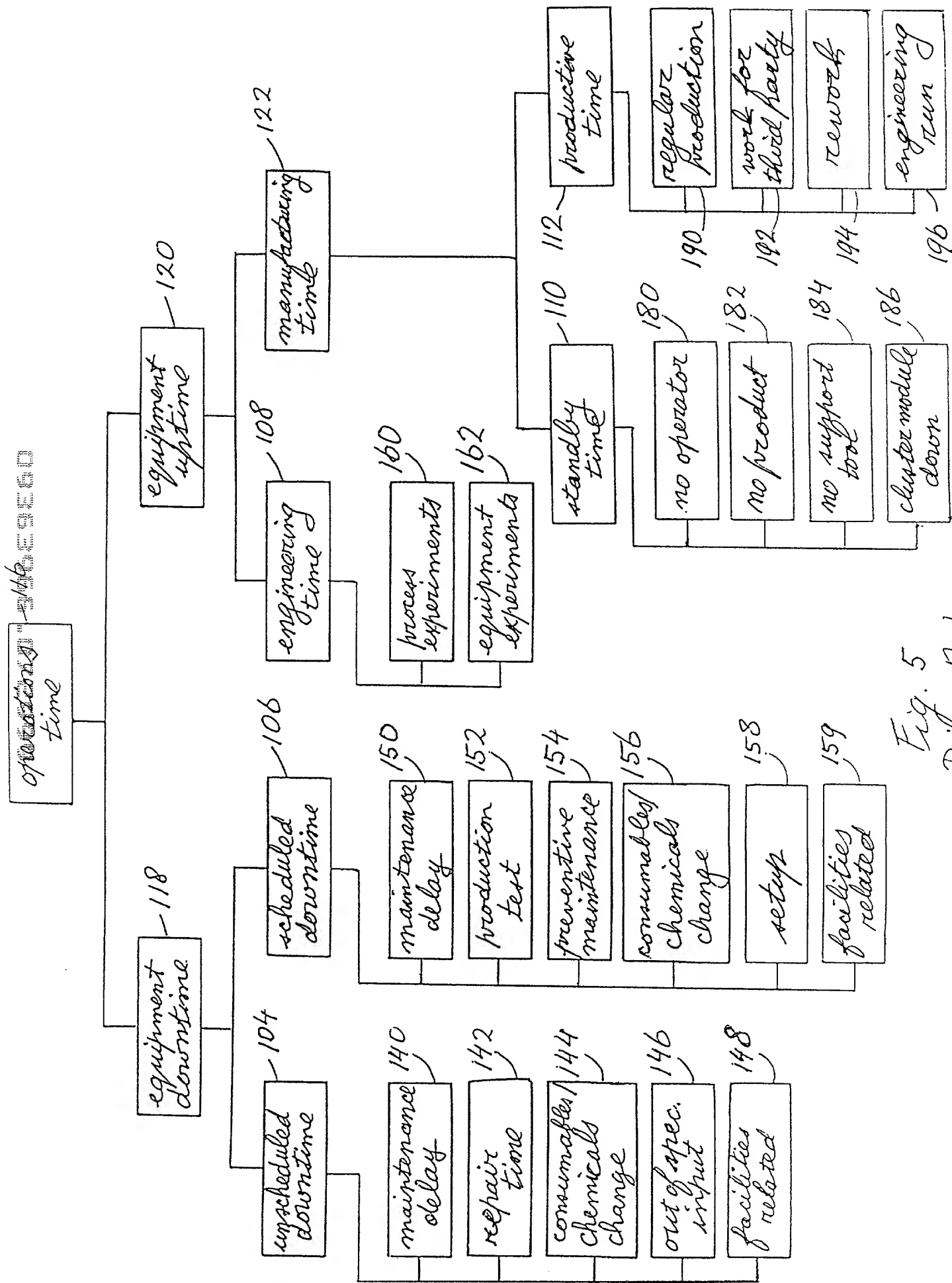


Fig. 5
Prior Art

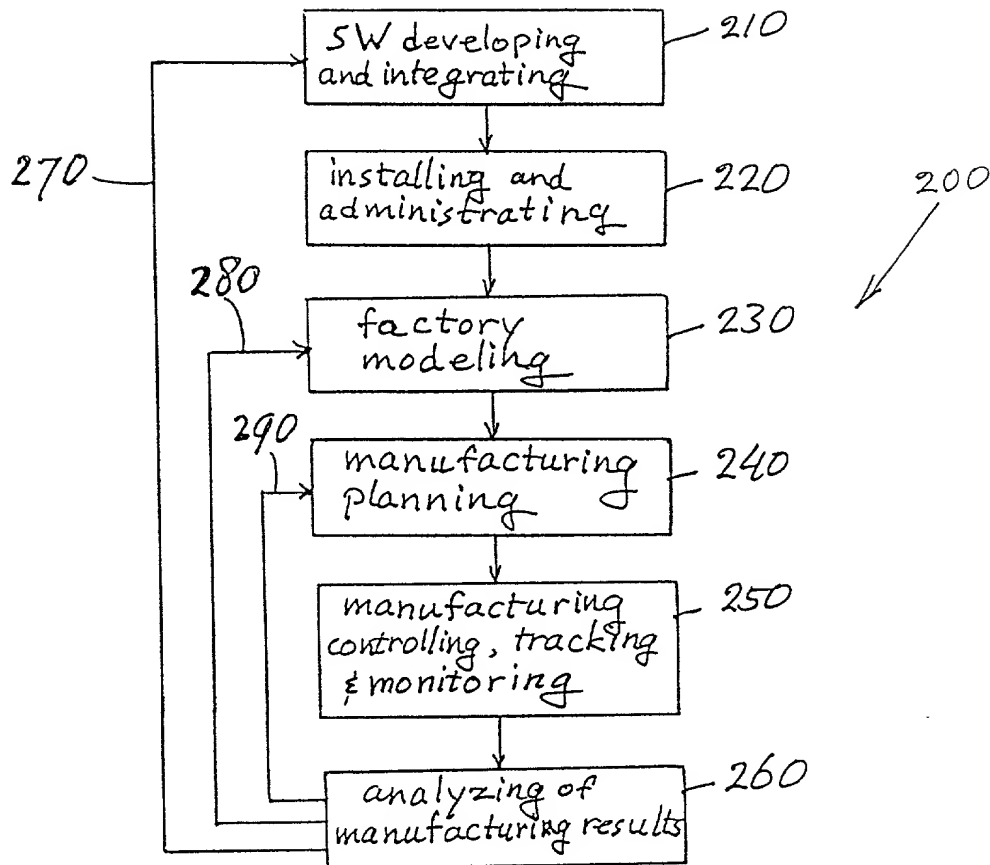


Fig. 6

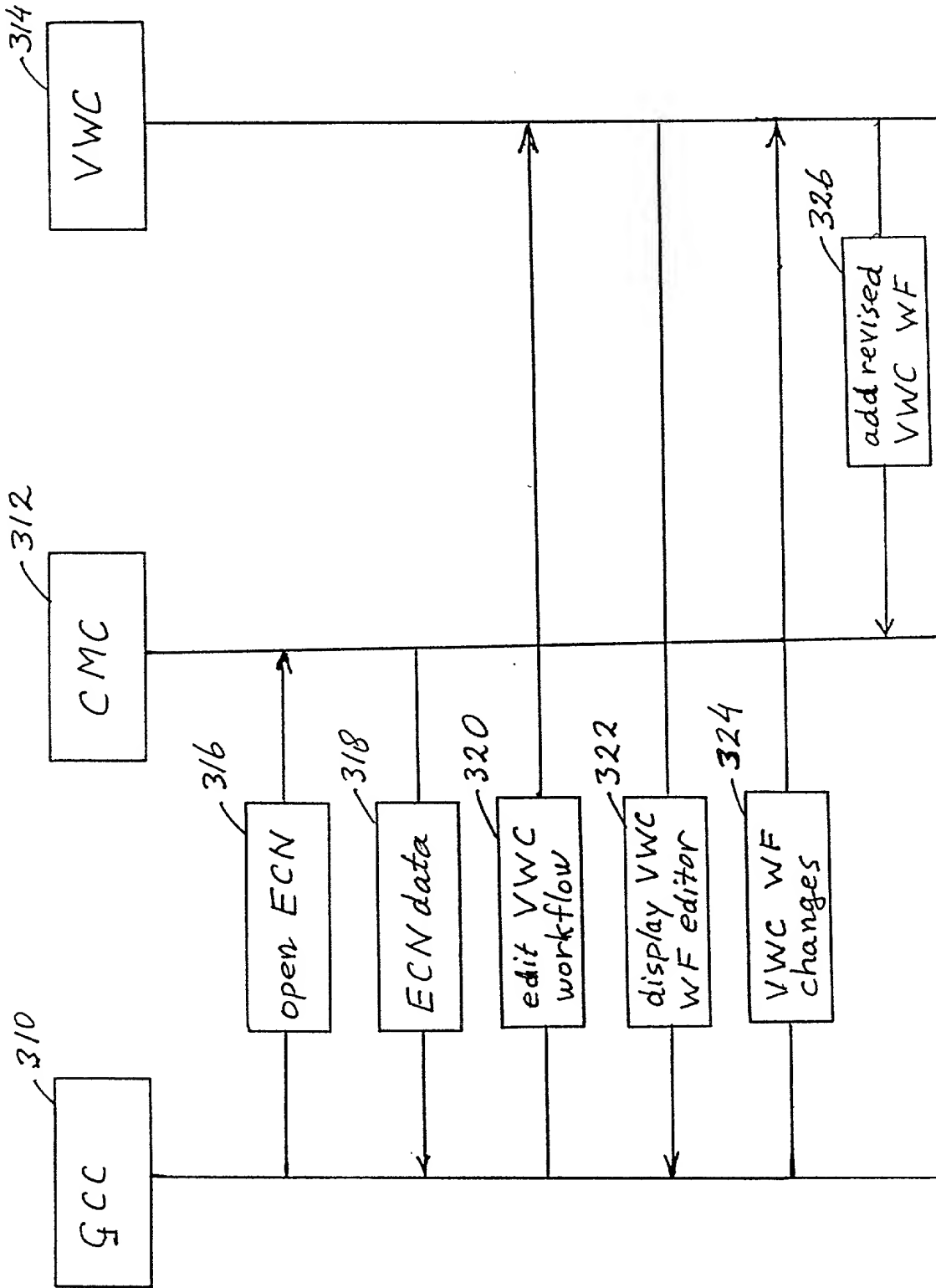


Fig. 7

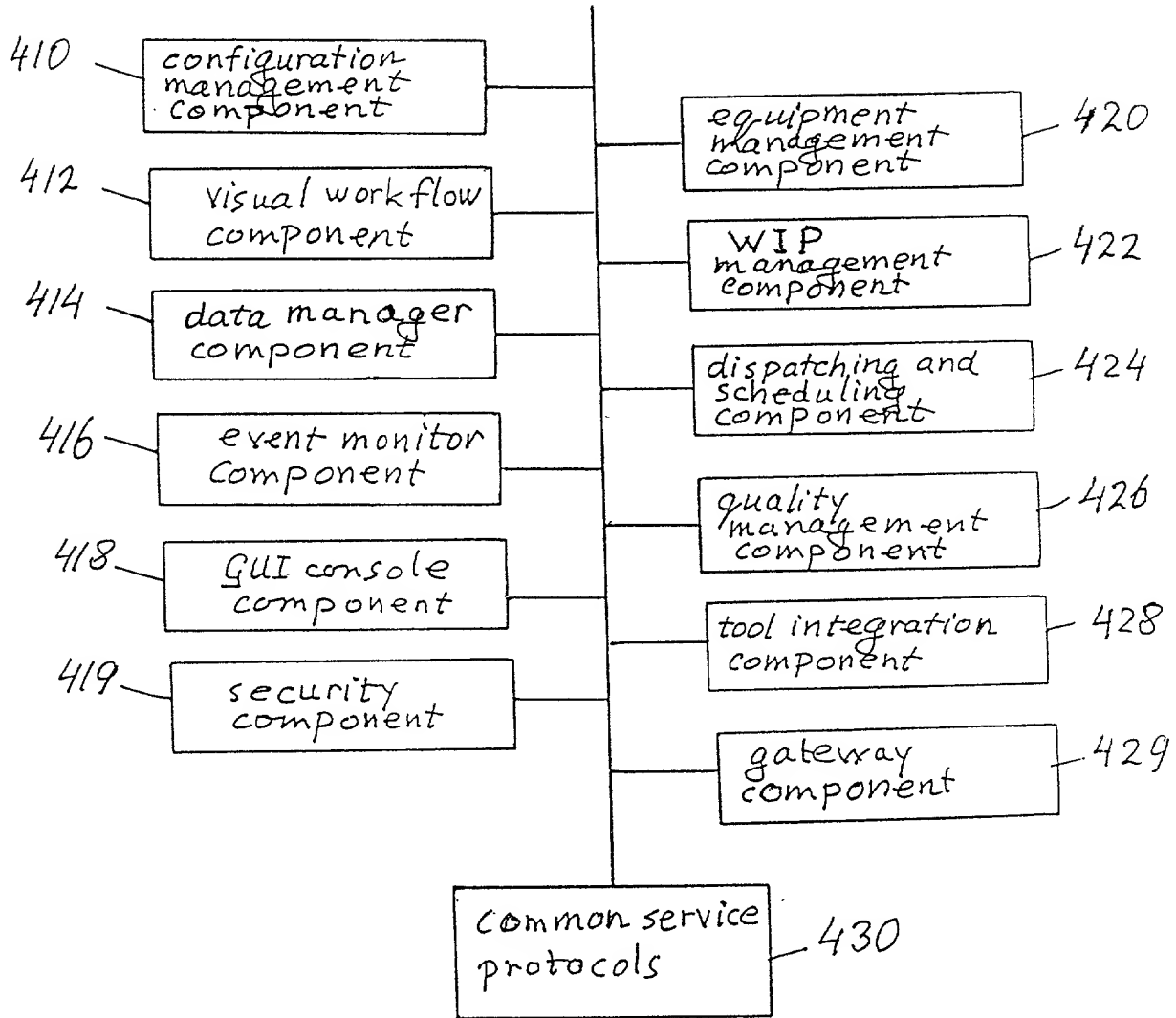


Fig. 8

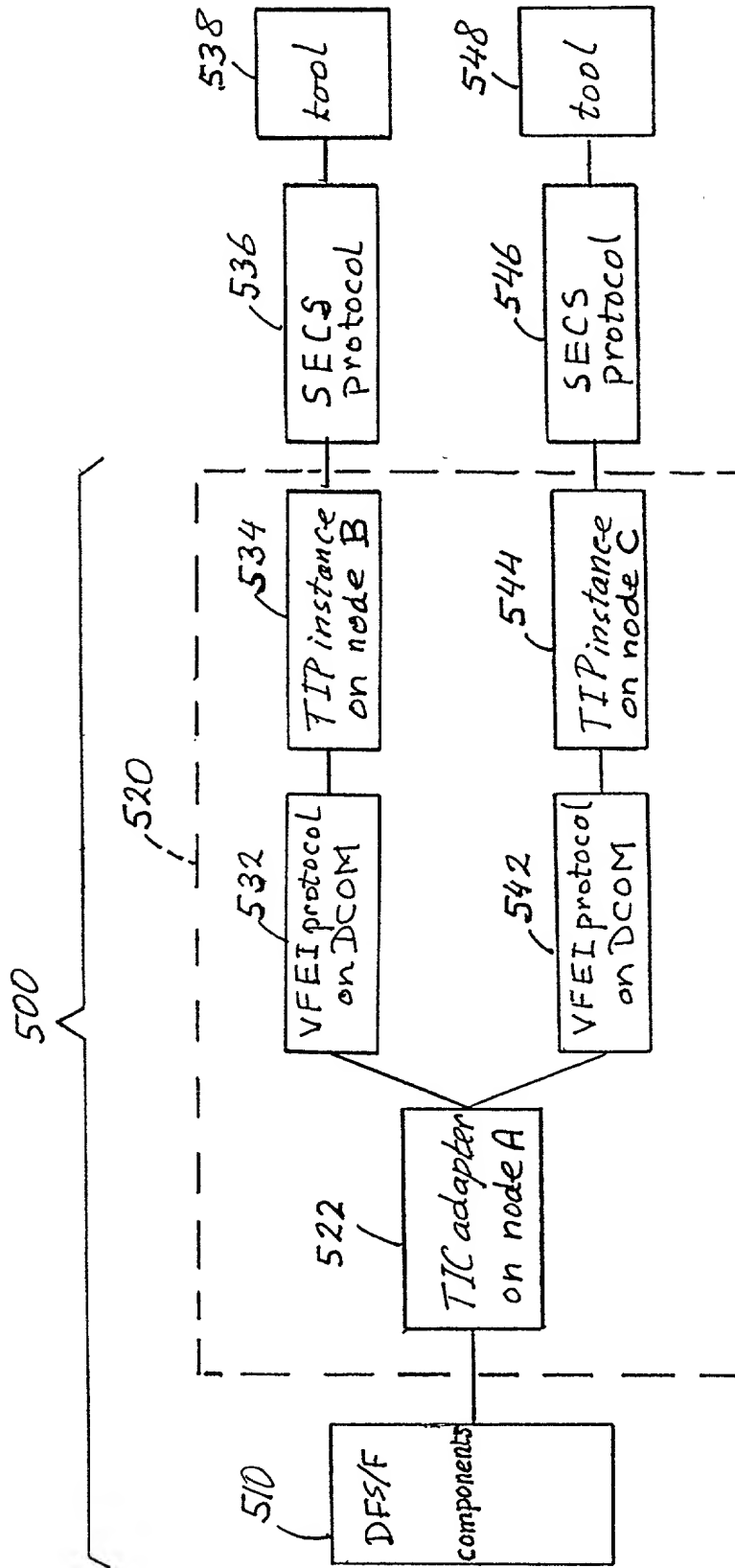


Fig. 9

656220" 99663860

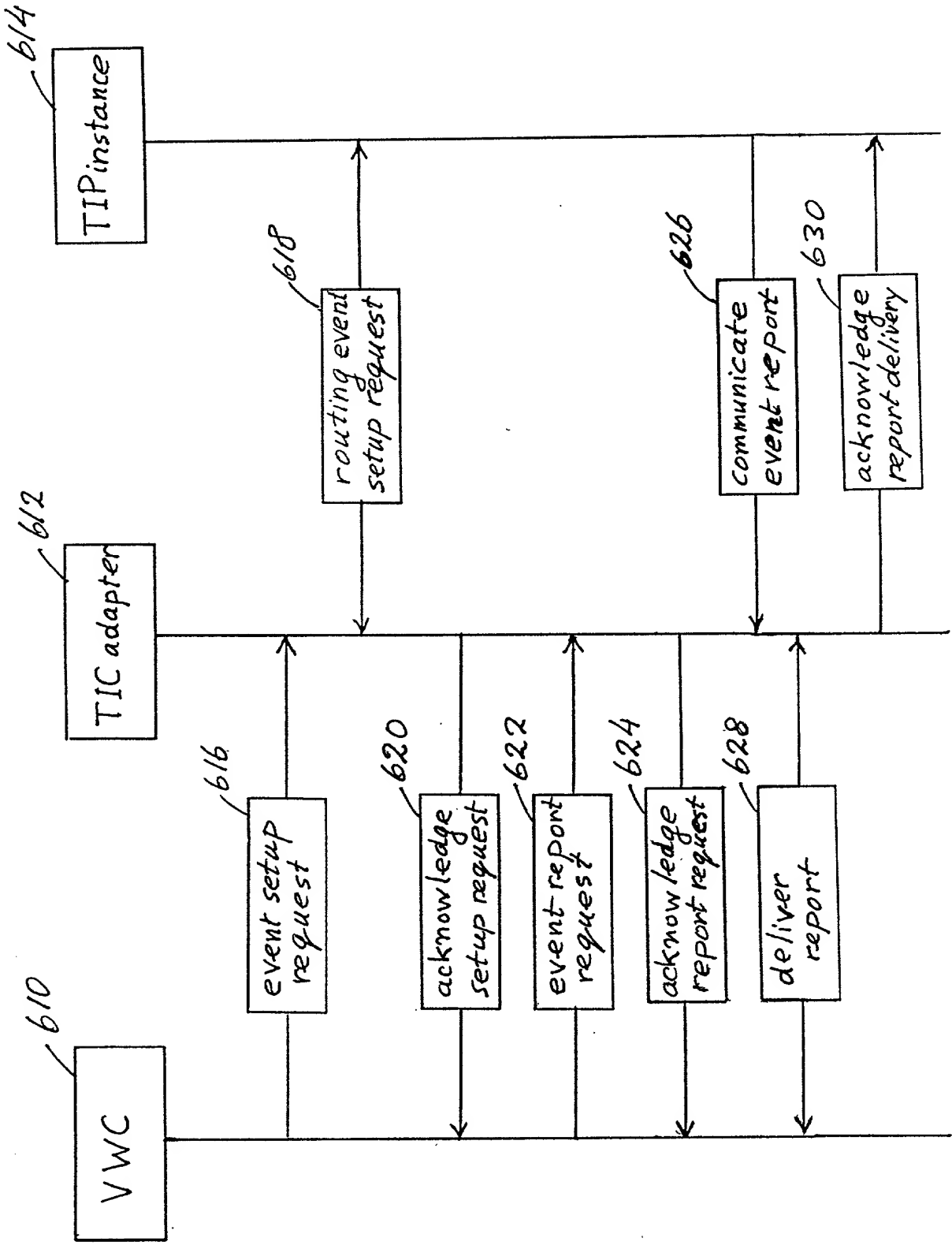


Fig. 10